

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-100148

(43)Date of publication of application : 02.04.1992

(51)Int.Cl.

G06F 9/06

G06F 15/00

(21)Application number : 02-263242

(71)Applicant : SUN MICROSYST INC

(22)Date of filing : 02.10.1990

(72)Inventor : CORBIN JOHN R

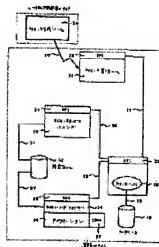
(30)Priority

Priority number : 89 415984 Priority date : 02.10.1989 Priority country : US

(54) METHOD FOR PREVENTING UNAUTHORIZED USE OF SOFTWARE INSIDE COMPUTER NETWORK SYSTEM

(57)Abstract:

PURPOSE: To prevent unauthorized use of a software application by confirming a compounded license token means before the certification of access to a license. CONSTITUTION: When a user desires the execution of the software application, a licensing library 24 performs calling for requesting a license token from a license server 20. The license server 20 detects the proper license token of the software application and transmits this license token to the licensing library 24, and a license access module connected to the licensing library 2 the license token. A routine inside the licensing library 26 barrows the license and confirms license information before the license token is updated. Thus, the illegal use of the software application is prevented.



⑫ 公開特許公報(A) 平4-100148

⑬ Int. Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成4年(1992)4月2日

G 06 F 9/06
15/00

4 5 0 P
3 3 0 B

7927-5B
7218-5L

審査請求 未請求 請求項の数 2 (全17頁)

⑮ 発明の名称 コンピュータネットワークシステム内でのソフトウェアの不正使用を防ぐ方法

⑯ 特 願 平2-263242

⑰ 出 願 平2(1990)10月2日

優先権主張 ⑱ 1989年10月2日 ⑲ 米国(US) ⑳ 415,984

㉑ 発 明 者 ジョン・リチャード・ アメリカ合衆国 94086 カリフォルニア州・サニヴエ
コービン イル・アメリカ アヴェニウ・372

㉒ 出 願 人 サン・マイクロシステムズ・インコーポレー アメリカ合衆国 94043 カリフォルニア州・マウンテン
テッド ビュー・ガルシア・アヴェニウ・2550

㉓ 代 理 人 弁理士 山川 政樹 外3名

明 和 書

1. 発明の名称

コンピュータネットワークシステム内でのソフトウェアの不正使用を防ぐ方法

2. 特許請求の範囲

111 少なくとも1個のネットワークを構成する複数の端末上で作動すべくライセンスされた複数のソフトウェアアプリケーションを含むコンピュータネットワークの環境において、前記アプリケーションは前記端末上にあり、特定の端末におけるアプリケーションの使用はライセンスの付与によって許可され、前記ライセンスは前記アプリケーションの存在する前記端末からユーザによって要求されることを特徴とするコンピュータネットワークの環境において、前記アプリケーションの不正使用を防ぐシステムであって、

前記アプリケーションのライセンシング情報を記憶するライセンストークン手段と、

前記端末に接続されて前記アプリケーションと

通信を行なうライセンスサーバ手段であって、前記ライセンストークン手段を記憶するデータベースを有し、前記アプリケーションによるライセンスへの要求に応じて前記データベースから前記ライセンストークン手段を検索し、さらに前記ライセンストークン手段を前記アプリケーションへと伝達するライセンスサーバ手段と、

前記端末に接続され、前記ライセンスサーバ手段から伝達されるライセンストークン手段の符号化と復号化を行なうライセンスアクセス手段であって、前記アプリケーションと一体化され、前記ライセンストークン手段を前記ライセンスサーバ手段から受領するライセンスアクセス手段と、

前記端末に接続され、前記ライセンスへのアクセスが認可される前に前記符号化されたライセンストークン手段を記憶するライセンシングライブラリ手段であって、前記アプリケーションと一体化されたライセンシングライブラリ手段と、からなることを特徴とする前記アプリケーションの不正使用を防ぐシステム。

(1) 少なくとも1個のネットワークを構成する複数の端末上で動作するライセンスされた複数のソフトウェアアプリケーションを含むコンピュータネットワークの環境であって、前記アプリケーションは前記端末上にあり、特定の端末におけるアプリケーションの統括はライセンスの付与によって認可され、前記ライセンスは前記アプリケーションの存在する前記端末からユーザによって要求されることを特徴とするコンピュータネットワークの環境において、前記アプリケーションの不正使用を防ぐシステムであって、

前記アプリケーションのライセンスング情報を記憶するライセンストークン手段と、

前記端末に接続されて前記アプリケーションと通信を行うライセンスサーバ手段であって、前記ライセンストークン手段を記憶するデータベースを有し、前記アプリケーションによるライセンスへの要求に応じて前記データベースから前記ライセンストークン手段を検索し、さらに前記ライセンストークン手段を前記アプリケーションへと

ス結合手段と、
からなることを特徴とする前記アプリケーションの不正使用を防ぐシステム。

3 発明の詳細な説明

〔産業上の利用分野〕

本発明は、コンピュータネットワーク環境におけるソフトウェアアプリケーションの不正使用を防ぐ方法に関する。

〔従来技術〕

コンピュータネットワークとは、一般にリンクやケーブルで相互に接続された複数のマシンまたは端末をいう。コンピュータネットワークは、アクセスに対してオープンであるという特徴を有するため、ソフトウェアを許可なく複製することが可能である。このためソフトウェア開発者のライセンスング収入が減少する結果となっている。従来、ライセンスによってネットワーク全体の実施の許諾を行なう（一般にサイトライセンスと呼ば

れる）ライセンスサーバ手段と、

前記アプリケーションに接続され、前記端末からアクセス可能であり、前記ライセンスサーバ手段から伝達されるライセンストークン手段の符号化と復号化を行なうライセンスアクセス手段であって、前記アプリケーションと一体化されたライセンスアクセス手段と、

前記アプリケーションに接続され、前記端末からアクセス可能であり、前記ライセンスへのアクセスが許可される前に前記復号化されたライセンストークン手段を複製するライセンスングライブラリ手段であって、前記アプリケーションと一体化されたライセンスングライブラリ手段と、

前記ライセンスサーバ手段と前記ライセンスングライブラリ手段とに接続され、結合ファイルを形成するライセンス結合手段であって、前記結合ファイルは前記ライセンスングライブラリ手段に附して前記ライセンスサーバ手段のうち、いずれの手段が前記アプリケーションにライセンスを付与できるかを通知することを特徴とするライセン

ス結合手段と、
からなることを特徴とする前記アプリケーションの不正使用を防ぐシステム。

る）か、もしくはソフトウェアが実行される各ノードをライセンスする（一般にノードライセンスと呼ばれる）必要があった。ノードとは、コンピュータネットワーク内の各マシン、端末もしくはシステムを指す。ライセンスとは、ソフトウェアの開発者が顧客に対して任意のソフトウェアアプリケーションを特定の環境で使用することを許諾することをいう。

サイトライセンスの場合、特定の場所すなわちネットワーク上の全ユーザに、各々のネットワーク上の位置に関係なくソフトウェアアプリケーションの使用を許可する。均一の使用料を徴収するこの方法は、使用頻度の低いソフトウェアアプリケーションの場合には行き過ぎとなる。ノードライセンスの場合、ソフトウェアアプリケーションをネットワーク上の特定のマシンに結びつけてしまうことになり、またソフトウェアアプリケーションの使用頻度が低ければ採算がとれない、これらの点については米国特許4,698,189に開示されている。またライセンスを付与されたノードで所

たなユーザがソフトウェアアプリケーションを
用しようとする場合、こうしたユーザはライセン
スの新規購入を求められることが多い。

サイトライセンスやノードライセンスに代わる
概念として並行使用ライセンスがある。並行使用
ライセンスでは、任意の時点でソフトウェアアプ
リケーションの使用を許可されるユーザの人数を、
ネットワーク上の装置に依りなく限定している。
ビデオ愛好者がレンタルビデオ店から映画のビデ
オを借り出すように、ネットワーク上のユーザは
ソフトウェアアプリケーションを端末から先着順
に借り出す。このように、並行使用ライセンスで
は、ソフトウェアアプリケーションの実際の使用
度に比例して使用料が徴収される。

ネットワーク環境でソフトウェアアプリケー
ションの並行使用を許可するというライセンシング
方法は、現在ハイランド・ソフトウェア社とアポ
ロ・コンピュータ社で採用されている。これに関
しては、「並行アクセスライセンシング(Concur
rent Access Licensing)」(1988年9月、Vol. 8、

No. 8、ユニックス・レビュー(Unix Review)、H.
オルソン、P.ルヴァイン(H. Olson and P. Levin
e)著)で述べられている。一般に、ソフトウェア
アプリケーションのライセンスは、ライセンスサ
ーバの制御するデータベースに記憶されている。
ライセンスサーバとは、ライセンスを記憶してい
るプログラムであり、かつライセンスを貸出する
にユーザの資格を確認するプログラムである。不
正使用を防ぐために並行使用をライセンスするこ
れらの方法は、公共/個人キー暗号化方式などに
よって保護された通信手段に依存している。かか
る公共/個人キー暗号化方式によれば、システム
の各ユーザは2個のキーを有する。そのうち1個の
キーは通常公共のキーであり、他の1個は個人用
のキーである。個人キーを使用した個人認証は、
公共キーを使用した公共認証に異な付けられてい
るが、個人キーは公共キーから計算によって求め
ることはできない。これらの点については、B.デ
ニング(B. Denning)著「暗号化技術とデータ保護
(Cryptography and Data Security)」(1982年、

アディソン・ウェスリー(Addison-Wesley)社刊)
に述べられている。暗号キーはライセンスサーバ
内に格納されていて、ライセンスのデータベースの
暗号化に使用される。高度な技術で設計された公
共/個人キー暗号化方式を解読することは難しい。
とくにライセンスサーバが安全な環境内にあれば、
かかる方式の解読は至難の技である。安全な環境
とは、そのアクセスが資格を有するユーザのみに
限定されている環境をいう。しかしながら、ライ
センスサーバは顧客のサイト、すなわち危険な環
境に設置されることが多い。したがって、ライ
センスサーバは熟練したハッカーの解読にさらさ
れることになる。いったん個人キーが解読され
ると、ライセンスサーバに関するあらゆる重要な情
報(ライセンス等)が無許可のまま公開されてしま
う。

したがって、本発明の目的は、並行使用ライ
センシングの環境におけるソフトウェアの不正使用
を防ぐためのより安全性な方法を提供することに
ある。

[発明の概要]

本発明は、通常ライセンスサーバが行なってい
た認証機能とライセンス貸出機能とをソフトウェア
アプリケーションに与えるものである。本発明
の好ましい実施例は、少なくとも1個のライセン
スサーバと少なくとも1個のソフトウェアアプリ
ケーションとを実行する複数の端末を含むコンピ
ュートネットワークからなる。ライセンスサーバ
は、ソフトウェアアプリケーションのライセンス
情報を格納している端末のデータベースを制御す
る。ライセンス情報はライセンストークンに含ま
れており、またライセンスサーバが制御するデー
タベースに記憶されている。ライセンストークン
は、特定のビットパターンすなわちパケットであ
って、アプリケーションソフトウェアの販売業者
によって暗号化されている。ソフトウェアアプリ
ケーションは、ライセンシングライブラリを介し
てライセンスサーバと通信を行なう。ライセン
シングライブラリはライブラリルーチンの集合であ
って、ソフトウェアアプリケーションがライセン

スをライセンスサーバから要求する場合か、もしくは該ライセンスを更新する場合にこれらのライブラリルーチンを使用する。ライセンスアクセスモジュールは、ソフトウェアアプリケーションとライセンスングライブラリとに接続しているプログラムであって、ライセンストークンを断続的に特定のフォーマットからライセンスングライブラリのフォーマットへと変換化するプログラムである。

ユーザがソフトウェアアプリケーションの実行を希望する場合、ライセンスングライブラリは、ライセンストークンをライセンスサーバから要求するための呼出しを行なう。従来、ライセンスサーバはユーザの資格を照会したのち該要求を許可または却下していたが、本発明のライセンスサーバは、当該ソフトウェアアプリケーションの正しいライセンストークンを検出してこれをライセンスングライブラリに伝達する。ライセンスングライブラリに伝達されたライセンスアクセスモジュールは該ライセンストークンを符号化する。ソ

フトウェアアプリケーションに結合されたライセンスングライブラリ内のルーチンは、ライセンスを借り出してライセンストークンを更新する前に、ライセンス情報を確認する。ライセンスアクセスモジュールは、更新されたライセンストークンを符号化したのち、該トークンをライセンスサーバへと返送する。

ライセンストークンの複製検出と借り出し機能とがソフトウェアアプリケーションによって実行されるため、ライセンスサーバに代わってソフトウェアアプリケーションが無許可ユーザの解放の対象となる。ライセンスアクセスモジュールにリパースエンジニアリングをかけたも、ライセンスサーバの解放より実りが少ない。この理由は、ライセンスアクセスモジュールはライセンスデータベースのごく一部の内容を表示するに過ぎないからである。大部分のハッカーがライセンスアクセスモジュールを解読する頃には、ソフトウェア販売者は当該ソフトウェアアプリケーションの新バージョンに、これに対応する新たなライセンス

アクセスモジュールを発売しているはずである。このように、本発明は、基本的なコンピュータネットワークを盗用することなく、コンピュータネットワーク環境におけるソフトウェアアプリケーションの不正使用を防ぐためのより安全な方法を提供する。

[表記と名称]

以下の詳細な説明では、データビットやコンピュータメモリ内のデータ構造に対して行なわれる演算のアルゴリズムや記号表記を主として使用する。アルゴリズムに基づくこれらの記述や表記は、データ処理分野の有能な技術者が、各々の作業の内容を他の読者にも最も効果的に伝達するために使用する手段である。

以下でいうアルゴリズムとは、通常定義されるように、所望の結果を得るための一貫性のある一連のステップをいう。これらのステップは、物理量の物理的な操作を必要とするステップである。これらの物理量は、必ずしも限定されないが、電

気的もしくは磁気的な信号の形態をとることが普通であり、かかる信号に対しては記憶、転送、組合せ、比較その他の操作を行なうことができる。主として慣例上の理由で、これらの信号をビットパターン、値、要素、記号、文字、データパッケージ等の名称で呼ぶことが便利であることが多い。しかしながら、これらの用語もしくはこれに換する用語は、適切な物理量に対応していなければならない。かつ該用語はこれらの数量に付加された便利なラベルに過ぎないことを示しておく必要がある。

さらに、これらの物理量に対する操作は、加算や比較といった用語で表現され、こうした用語は人間が自分自身で行なう演算に通常関連付けられている。しかしながら、以下で説明する本発明の一部をなす演算では、人間のかかる能力はほとんどの場合不用であり、むしろ好ましくない。演算はマシンが実行するからである。本発明は、汎用デジタルコンピュータもしくは同等の装置に適用して効果的である。いずれの場合でも、コンピ

ュータを操作する方法と、計算自体の方法とを明確に区別する必要がある。本発明は、電気信号もしくは他の物理的(機械的、化学的など)信号を処理するためにコンピュータを操作することによって、他の形態の物理的信号を生成するスタップの方法に関する。

また本発明は、上記の操作を実行する装置に関する。かかる装置は、所望の目的のために形成されるか、もしくは汎用コンピュータからなり、該コンピュータ内に記憶されたコンピュータプログラムによって選択的に起動され、再構成される。本明細書に含まれるアルゴリズムは、特定のコンピュータやその他の装置に限定されるものではない。また各種の汎用マシンを、本明細書記載の命令に準じて記述されたプログラムとともに使用してもよい。所定のスタップを実行するための専用の装置を形成することが便利である場合もある。これらの各種マシンの所定の構造は、以下の説明から明瞭になるであろう。

支えない。好ましい端末の例としては、サン・マイクロシステムズ(Sun Microsystems)社(在カリフォルニア州マウンテンビュー(Mountain View))の製造するマシンがある。各端末は、キーボード11、11'、11''またはマウス12、12'、12''などの入力装置を有する。第1図に示されるように、端末10-10''(図中では10、10'、10'')は共通ケーブル12によって接続され、相互にデータの転送を行なう。共通ケーブル12は、両端ケーブル、光ケーブル、無線チャネルその他のいかなる共有メディアでもよいことは当業者には明白であろう。さらにケーブル12と端末10-10''(図中では10、11'、10'')からなるネットワークは、環状結線、星形結線、バス結線など多岐の形態のうちいずれをもとることができ、またゲートウェイやブリッジで結合された複数の小形のネットワークの集合を含んでいてもよい。

再び第1図において、14はライセンスサービスを示す。ライセンスサービス14は、ネットワークに接続されたすべての端末が共有する資源である。

[発明の実施例]

以下の説明は複数の図に分割されている。最初の図では、ライセンスされたソフトウェアプログラムのデータベース14にアクセスするための汎用ネットワーク環境を説明する。続く図では、ソフトウェアアプリケーションの不正使用を防ぐ方法の詳細を説明する。

1. 汎用ネットワーク環境

第1図において、コンピュータネットワーク環境は、複数のデータ処理装置10-10''(図中では10、10'、10'')からなる。これらのデータ処理装置には、端末、パソコン、ワークステーション、ミニコン、メインフレームコンピュータ、スーパーコンピュータ等が含まれる。本明細書では、本発明のネットワークに接続されたすべてのデータ処理装置を「端末」と称する。これらの端末の製造元は異なるものであっても構わない。また該端末の使用するオペレーティングシステムは、MS-DOS、UNIX、OS/2、MAC OS等さまざまであって差

本実施例では、ライセンスサービス14にはライセンスサーバ15-15''(図中では15、15'、15'')とデータベース17-17''(図中では17、17'、17'')が含まれる。ここで、nはn以下の値である。ライセンスサーバは、メモリ型記憶装置を有する端末上で実行されるプログラムである。各ライセンスサーバ15(図中では15、15'、15'')は、インタフェース18(図中では18、18'、18'')を介して、端末のメモリに記憶されたデータベース17と通信を行なう。以下で詳しく説明するように、データベース17は、各種のソフトウェアアプリケーションのライセンシング情報を記憶している。これらのソフトウェアアプリケーションは、購入後、コンピュータネットワーク環境で実行するための許可を付与される。ライセンスサーバは、特定の端末での実行に限定されるわけではなく、いずれの端末でも実行可能であり、ユーザがアプリケーションを使用する端末でも実行できる。このように、ネットワークに接続された端末であれば、ユーザがアプリケーションソフトウェアを実行する端末

としてだけでなく、ライセンスサーバとしても機能できる。以下で詳しく説明するように、ライセンスサーバはアプリケーションソフトウェアのライセンスの管理を行なわない。ライセンスサーバの作動形態は受動的であって、記憶装置、ロック機能、ログ機能およびクラッシュ回復機能をアプリケーションソフトウェアに提供している。

第2図は、本発明に基づくネットワークライセンシング方式の構造を示す。この構造には、データベース18、データベースインタフェース19、ライセンスサーバ20、ライセンシングライブラリ24、ライセンスアクセスモジュール27、ライセンス管理ツール21、ライセンスサービスバインディング28、ライセンス生成ツール34が含まれる。

データベース18は、ライセンシング情報とアプリケーション使用データを記憶している。データベース18には、以下の情報を含む複数のレコードが格納されていることが望ましい。

データベースの要素

・ 一意キーテーブル	他の全テーブル用キー
・ 販売業者テーブル	販売業者の識別記号(10)と名前
・ 製品テーブル	製品の番号と名前
・ バージョンテーブル	バージョンの番号と日付
・ ライセンステーブル	ライセンス番号、失効日、総数
・ ライセンストークンテーブル	符号化されたライセンストークンを格納するテーブル
・ 単位グループテーブル	任意のグループ内のライセンスの配分
・ グループリストテーブル	グループの名前
・ 使用許可ユーザのテーブル	使用許可ユーザの資格
・ 実行ライセンス使用テーブル	ライセンスを使用中のアプリケーション

データベースの要素

・ ロックテーブル	データベース中のロックされたレコード
・ 管理者管理テーブル	管理者のログイン名
・ ライセンス操作ログテーブル	管理者のログ情報
・ ライセンス使用ログテーブル	要求操作とクライアントログ
・ ライセンス待ち行列ログテーブル	ライセンス待ち行列
・ アプリケーションメッセージログテーブル	アプリケーションに與有なメッセージ

データベースインタフェース19は、ライセンスサーバ20とデータベース18との間の通信を提供する。これによって複数のユーザがデータベース内の同一レコードに同時にアクセスすることを防いでいる。かかる同時アクセスが行なわれると、該レコード内のデータが破壊されることがある。

このように、アプリケーションの使用時にはロックの所有権のみがロックされたレコードからデータを読み出し、書き込むことができる。

ライセンスサーバ20は端末上で作動し、データベース18と、ライセンス管理ツール21、ライセンシングライブラリ24およびライセンスサービスバインディング28とのインタフェースをとる。ライセンスサーバ20は、インタフェース23を介して、ライセンス管理ツール21、ライセンシングライブラリ24およびライセンスサービスバインディング28と通信を行なう。インタフェース23は、遠隔手段呼出し機構であることが望ましい。遠隔手段呼出し機構によれば、ネットワークに接続された1個の装置または端末上で作動するプロセスは、該ネットワークに接続された遠隔装置または遠隔端末から資源またはサービスを要求できる。この点に関しては、A.ピレル、B.ネルソン(A. Birrell and B. Nelson)著「遠隔手段呼出しの実施(Implementing Remote Procedure Calls)」(1984年 Vol. 2, No. 1, エンシーエム・トラニング・システム・オブ・コンピュータ)

ュータ・システムズ(ACH Transaction on Computer Systems)記載)に述べられている。

複数のライセンスサーバが複数の端末に常駐することができる。ライセンスサーバ20は、端末のバックグラウンドモードで動作することによって、該端末を使用するユーザからみて透明であることが望ましい。また以下で説明するように、ライセンスサーバ20は次のような機能を提供する。

- (1) ライセンシングライブラリからなされるライセンストークンの要求を処理する機能
 - (2) いずれのライセンシング単位も使用できない時、データベース18への要求で構成される待ち行列を管理する機能
 - (3) データベース18への並行的アクセスのロックを生成する機能
 - (4) データベース18内の情報にアクセスする機能
- ライセンシングライブラリ24は、ライブラリルーチンの集合であり、アプリケーション28はこれらのルーチンによってライセンスサーバ20にライセンシングサービスを要求する。ライセンスサー

バ20は、ライセンシングライブラリ24からサービスの要求を受信すると、データベース18からライセンストークン₂₄を検索して該トークンをライセンシングライブラリ24へと返却する。ライセンシングライブラリ24はアプリケーション28と接続しており、経路28を介してライセンスサーバ20と通信を行なう。かかる通信は、通常手続呼び出し方式で行なわれることが望ましい。ライセンシングライブラリ24における主要なライブラリ呼出しには、アプリケーションがライセンスサーバ20に対して行なうライセンスの要求が含まれる。その他、ライセンスの更新要求と解除要求が重要なライブラリ呼出しの一部である。ライセンストークンを使用して各種のライセンシングサービスの要求を行なう方法を以下に説明する。

ライセンスアクセスモジュール(LAM)27は、ソフトウェアの販売業者が用意するものであって、ライセンストークンの復号化を目的としている。アプリケーション28は、復号化されると、ライセンシングライブラリ内のルーチンを使用してライ

センストークン内のライセンシング情報を確認し、ライセンスが貸出されてよいかを決定する。またライセンスアクセスモジュール(LAM)27は、アプリケーションがライセンスサーバ20を介してライセンストークンをデータベース18に戻す前に、該ライセンストークンを符号化している。以下ではライセンスアクセスモジュール27をさらに詳細に説明する。

ライセンス管理ツール21は、ネットワーク管理者がソフトウェアアプリケーションの並行使用に関する管理機能を実行する場合に使用する。ライセンス管理ツール21は、コンピュータネットワークに接続された端末であればいずれの端末でも実行可能である。ライセンス管理ツール21は、ライセンストークンをライセンスサーバ20を介してデータベース18にインストールする場合に主として使用される。ライセンス管理ツール21には次の機能が含まれる。

- (1) ライセンスサーバを起動、停止する機能
- (2) ライセンスサーバの制御するデータベースに

アクセスする機能

- (3) ライセンスの使用に関する報告を生成し、印する機能

アプリケーション28はデータベース18を直接アクセスすることはできない。このアクセスを達成するためには、アプリケーションはライセンシングライブラリ24から経路28を介してライセンスサーバ20宛てにライセンスの要求を行なう。大部分のネットワークライセンシング方式では、ライセンシングライブラリ24とライセンスサーバ20との間に機密保護された伝送路を採用している。しかしながら本発明では、ライセンスアクセスモジュール(LAM)27、ライセンスライブラリ24および複数のライセンストークンを使用してコンピュータネットワークにおけるソフトウェアアプリケーションの不正使用を防止している。

再び図2において、ライセンスサービスバイнда29は経路30を介してライセンスサーバ20と接続されている。このライセンスサービスバイнда29は、ネットワークサービスプログラム等の本分

野の公知技術によって呼出される。ライセンスサービスバイナリ29は、ネットワーク上でサーバとして指定されたすべての端末の位置を検出し、いずれのサーバがいずれのアプリケーションの処理を行なっているかを記録し続ける。ライセンスサービスバイナリ29は、自己の保有する可用サーバテーブルを参照して各サーバと連絡をとり、サーバの処理している製品のリストを各サーバに要求する。最後に、ライセンスサービスバイナリ29は、可用ライセンスサーバテーブルの内容と製品リストを、経路32を介して、結合ファイル32に書き込む。第2図において、結合ファイル32は経路32を介してライセンシングライブラリ24に接続されている。アプリケーション28は結合ファイル32に問い合わせを行ない、いずれのライセンスサーバが該アプリケーションのライセンスの要求を処理できるかを照会する。ライセンス生成ツール34は、ネットワーク管理者に伝達されるライセンストークンを生成するためにソフトウェア販売業者が使用する。ネットワーク管理者は、ライセンストークンを受

信すると、ライセンスサーバ20を介してライセンス管理ツール21によって該トークンをデータベース48へとインストールする。

11. ライセンストークン

本発明の好ましい実施例を使用してコンピュータネットワークで、ライセンストークンを生成する方法を第3図を参照しながら説明する。第3図において、コンピュータネットワーク38は、ライセンス管理ツール39と1個のライセンスサーバ44とに接続されている。ライセンスサーバ44はデータベース45と通信を行なう。アプリケーション41、42、43はライセンスサーバ44からライセンシングサービスを要求している。顧客が任意のアプリケーション(例: 研究開発部門用のCAD/CAMプログラム)のライセンスを購入すると、ソフトウェア販売業者はライセンス生成ツールによってライセンストークンを生成して、該トークンを顧客のネットワーク管理者に渡す。ライセンストークンは、ソフトウェアアプリケーションを使用するためのライ

センスを表現する特定のビットパターンすなわちパケットである。ネットワーク管理者は、ライセンストークン46を、ライセンス管理ツール39を使用してライセンスサーバのデータベースへとインストールする。トークンリング方式で端末から端末へと転送されるトークンと異なり、本実施例のライセンストークンは、一定の時間中ライセンスサーバとライセンシングライブラリとの間でのみ転送される。かかる一定期間は、ライセンストークンがライセンスサーバから貸出される時間の長さに相当する。現在、ライセンストークンはアプリケーションに対して10秒以内に転送され、該トークンはこれを貸出したライセンスサーバへと最短時間で返還される。ライセンストークン46は、販売業者独自のフォーマットで暗号化された情報を含んでいる。かかる情報には、販売業者の識別記号、製品番号、バージョン番号、ライセンストークンに対して購入されたライセンス単位の数などが含まれる。ライセンス単位は、コンピュータネットワークに接続された端末に加重されたライ

センスに対応している。たとえば、任意のソフトウェアアプリケーションを使用する場合、強力なワークステーションは平均的なパソコンより多くのライセンス単位を必要とすることになる。

ソフトウェア販売業者は、ライセンス生成ツール40を使用してライセンストークンを生成する。経路47は、顧客のサイトでライセンストークン46がどのようにライセンス管理ツール39へと送られるかを示している。ここでは、システム管理者はライセンストークン46をライセンストークン48として、ライセンスサーバ44のライセンスデータベース45へとインストールする。経路48は、ライセンストークン45がライセンストークン46としてライセンス管理ツール39からライセンスサーバ44へと転送される様子を示している。これにより、ライセンスサーバ44においては、トークン46に該当するアプリケーションを使用するライセンスの要求や、データベース45で示される他のアプリケーションを使用するライセンスの要求を、アプリケーション41、42、43から受け付ける用意が

完了する。

各ネットワークは複数のライセンスサーバを備
有可能であり、また各ライセンスサーバは、そのデ
ータベース内に複数のソフトウェアアプリケーション
に対応する複数のライセンストークンを含ん
でいてよいことは明らかである。再び第3図にお
いて、アプリケーションA 41が10秒未満でライセ
ンストークン48を要求してこれを借り出すと、ア
プリケーションA 41がライセンスをライセンスト
ークン48から借り出している期間中にアプリケー
ションB 42、C 43がこれに対する要求を行なっ
ても、ライセンストークン48を借り出すことが出来
ない。これはデータベースインタフェース19がロ
ック機構を提供しているためである。このように、
ネットワーク38でライセンスの旅行使用を達成す
るためには、ネットワーク管理者が2個以上のライ
センスサーバをインストールすることが望まし
い。ライセンスサーバのクラッシュから回復する
ための努力を軽減するためには、システム管理者
が各アプリケーションのライセンス単位を継続的

に配置された複数のライセンスサーバに分散して
おくことが望ましい。たとえば、ネットワークに
4個のライセンスサーバがある場合、使用頻度の
高いあるアプリケーションに対して2個のライセ
ンス単位があるとすれば、ネットワーク管理者は
4個のライセンストークンの各々を2個のライセン
ス単位を配分することが好ましい。1個のライセ
ンスサーバがクラッシュした場合、もしくは該サ
ーバのライセンストークンが買出されている場合、
他の3個のライセンスサーバが他のアプリケーシ
ョンに対してライセンシング処理を行なうことが
できる。

第4図は、ライセンストークンを使用してライ
センスを要求する方法を示している。第4図にお
いて、ネットワーク50がアプリケーション52、54、
56にそれぞれ結合されている。アプリケーション
58は、ステップ59においてライセンスサーバ58か
らライセンストークンを要求し、該要求は受け付
けられる。ステップ60において、ライセンストー
クンはアプリケーション58へと伝達される。続い

てアプリケーション58は、ステップ61においてラ
イセンストークンをライセンスサーバ58に返り返
す。第4図に示されるようなライセンストークン
によるライセンス要求過程に加えて、ライセンシ
ング処理の他の重要な段階でもライセンストー
クンが使用される。たとえば、ユーザが当初割り
当てられた時間を超過してアプリケーションを使
用したい場合がある。この場合、第4図に示される
ように、アプリケーション68は、ライセンストー
クン72を使用してライセンスサーバ70に対してラ
イセンス更新要求を発行する。同様に、第4図に
示されるように、ユーザはアプリケーションがラ
イセンス単位を必要としなくなった時点でライセ
ンス解除要求82を発行する。このように、ユーザ
はステップ85において、更新されたライセンスト
ークンをライセンスサーバ82に返すことによってラ
イセンストークン84を更新する。

111. ライセンスアクセスモジュール

第2図において、ライセンスアクセスモジュール

ル(LAN)27は、アプリケーション26とライセンシ
ングライブラリ24とに結合されて実行可能コード
を形成する。該コードは、ソフトウェア販売業者
が顧客に提供するコードである。ライセンスア
クセスモジュール27は、暗号化されたライセンス
トークンがライセンスサーバとライセンシングライ
ブラリ24との間で伝達される時点でその復号化と
符号化を行なう。このように、アプリケーション
の不正使用に対する機密保護のレベルは、ライ
センスアクセスモジュールの機密保護の度合いに大
きく依存する。

従来のネットワークライセンシングの方法では、
公共/個人キー暗号化方式を使用して重要な情報
を符号化している。この方式は、ライセンスサ
ーバが安全な環境内にある場合には効果的である。
しかしながら、顧客はライセンスサーバも含めた
ネットワークの全端末に同じようにアクセスする
ことができる。ユーザがライセンスサーバの個人
キーを解読すれば、ライセンシング方式の機密保
護は壊れてしまう。無資格のユーザがサーバの

個人キーをいったん特定すれば、ライセンスサーバに関する他のすべての重要な情報が解読されてしまう。仮にすべてのライセンスサーバが同一のキーを使用している（これがしばしば行われている）とすれば、すべてのライセンスサーバの各アプリケーションのすべての秘密保護が破られてしまう。

ライセンスアクセスモジュール27は、ライセンストークンを販売業者指定のフォーマットからライセンシングライブラリ24で使用可能なフォーマットへとまず変換する。ライセンスアクセスモジュールによるこの変換は2個のモジュールによって行われる。第一のモジュールは、ライセンストークンを販売業者指定のフォーマットからライセンシングライブラリのフォーマットへと変換すなわち符号化する。第二のモジュールは、更新されたライセンストークンをライセンシングライブラリのフォーマットから販売業者指定のフォーマットへと変換すなわち符号化する。第二のモジュールは、ライセンシングライブラリがライセン

ストークン内の情報を更新するたびごとに呼出される。

ライセンシングライブラリフォーマットに変換されたライセンストークンの受領後、ライセンシングライブラリはライセンスの正当性を確認するためのルーチンを実行する。該確認はトークン内に格納された次のようなライセンス情報を検討することによって行われる。

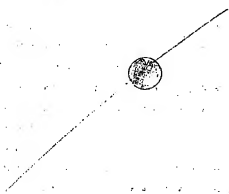
- (1) フラグ
- (2) 保守契約の日付
- (3) ホストの名称と定義域
- (4) 製品名
- (5) ホストの識別番号
- (6) ライセンスの連立番号
- (7) ライセンスの失効の日付

以上の情報がアプリケーションの保有する情報と比較される。これらの情報が合致すれば、ライセンスの確認が完了する。確認過程の終了後、ライセンシングライブラリ内でルーチンが起動される。このルーチンは、ライセンストークン内のラ

イセンス単位数から買出中のライセンス単位の数を減算してライセンスを算出す。

符号化ルーチンと符号化ルーチンとによって、ソフトウェア販売業者は、顧客のサイトにインストールするライセンスを不正使用から守るための独自の秘密保護機構を構築することができる。

以下に、ライセンシングライブラリとライセンスアクセスモジュールを使用するアプリケーションであって、図1で記述されたアプリケーションの例を示す。



```

5  #define LIC_RENEWAL_TIME 600 //set renewal time for this session/
  #define EST_LIC_RENEWAL_TIME (LIC_RENEWAL_TIME x 5)

  //set vendor #/
  //set vendor #/
  //set version id #/
  10  NL_vendor_id NL_Vendor_Id = 1232;
  NL_prod_num NL_Prod_Num = 1234;
  NL_version NL_Version = {123068, "1.0"};

  //initialise license service/
  //accept job id if no error/
  status = NL_init (Vendor_Id, NULL, App_Id);
  if (status != NL_NO_ERROR) //accept job id if no error/

  15  {
    //set (sidew, "id init failed - error =
    "%d", status);

    //error message if error and
    return;
  }

  20  code = NL_encode_p = nl_encode; //pointer to encode function/
  //NL_decode_p = nl_decode; //pointer to decode function/
  if (strcmp (SIGALRM, alarm_jml) == (void *) 1) //set alarm if no
    error;

  25  {
    //error ("Cannot set SIGALRM"); //otherwise, error message/
    return;
  }

  30  status = NL_request (Job_Id, NL_Prod_Num, //request a license/
    NL_Vendor_Id, NL_Version,
    EST_LIC_RENEWAL_TIME, NL_L2_SRCH,
    &code_func, NULL,
    &code_func, NULL, &app_jml);

  35  if (status != NL_NO_ERROR)
    //no error, license checked
    out from license server/
    {
      //print (sidew, "request failed - error =
      "%d", status);
      //otherwise, error message/
      return;
    }

  40  /* We got a license
  */
  //license request successful/

  45  alarm (EST_LIC_RENEWAL_TIME); //set alarm for license renewal
  time/
  //new application/
  //request to release a license/
  status = NL_release (code_func, &code_func);
  if (status != NL_NO_ERROR)

  50  {
    //print (sidew, "license failed - error =
    //otherwise, error

```

```

//error message/
return;
}

//new function with
//error message/
//error message/

5  //alarm_jml()

status = NL_confirm (req_handle,
  LIC_RENEWAL_TIME, NULL);
//Verify vendor private information

10  //

}

//otherwise, error
//error message/
//successful license
//renewal/

15  if (status != NL_NO_ERROR)
    //print (sidew, "id confirm failed - error =
    "%d", status);
    {
      //error ("license renewed")
    }

20  //

```

上記のアプリケーションの例では、コードの右側に示す関数が呼び出されている。特に注目すべきコードは、code_func.encode_pにcode_func.decode_pである。encode_pはdecode_pは、ソフトウェア版の署名の符号化ルーチンと復号化ルーチンにそれぞれ対応するポイントである。code_func関数のポイント例によると、ライセンスングライブラリは署名ポイントを使用してライセンスアクセスモジュール内の復号化ルーチンに符号化ルーチンと呼出している。3つの主要なライセンスングライブラリルーチン、すなわちライセンス要求ルーチン(NL_request)、ライセンス解除ルーチン(NL_release)およびライセンス更新ルーチン(NL_confirm)は、上記の復号化ルーチンと符号化ルーチンと呼出す。ライセンスアクセスモジュールの例については付録1を参照されたい。

ライセンスアクセスモジュールの実装において、ライセンスサーバはライセンスストリークの受け渡しとなるに適さない。ライセンスの付与に基づくアプリケーションの実行に先立って、アプリケーション

本発明は、前述のように第1-4図と付図1を参照しながら具体的に説明され、とりわけコンピュータネットワーク環境におけるソフトウェアアプリケーションの不正使用を防ぐ方法の実施について具体的に説明されたが、かかる説明は例であって本発明を限定するものではない。また本発明による方法は、コンピュータネットワーク環境で実行

されるいかなるアプリケーションに用いても効果的であることは明白である。さらに、以上の説明から本発明の精神と範圍とにそむくことなくさまざまな変更や修正を実行できることは当該分野の有識な技能者によって了承されよう。

-282-

明細書の注釈(内容に変更なし)

```

* Returns: int success
*   - Decoding succeeded
*   - HL_E_DECODE_FAIL
* Side effects:
*   - Allocates memory for the client usage entries.
* Notes:
*   - If tok->vend_ptr is NULL or tok->vend_ptr->data_p is NULL
*     then the call doesn't work.
* .....
```

```

static int
nl_decode(en_tok_p, en_length, tok_p, client_entry_p)
    en_tok_p *en_tok;
    int en_length;
    nl_token *tok_p;
    nl_client_entry **client_entry_p;
{
    nl_usage_entry *usage_entry;
    nl_client_entry *tmp_client_entry_p;
    int i;
    int status;
    int vend_size;

    /* Make sure the size for the encoded token is divisible by
     * 4 bits bytes per unit.
     */
    if (en_length % 4 != 0)
        return (HL_E_DECODE_FAIL);

    /* Init an XOR stream
     */
    xorstream_create(&addr, en_tok_p, (en_inten_length, XOR_DECODE));
    status = HL_NO_ERROR;
    if (addr.HL_license(addr, &n_l_license) == FAILURE)
        return (HL_E_DECODE_FAIL);

    if (n_license_magic == MY_VENDOR_MAGIC)
    {
        tok_p->vendor_id = n_license.tok.vendor_id;
        (void) strncpy(tok_p->prod_name, n_license.tok.prod_name,
            tok_p->max_name - n_license.tok.vendor_name);
        strncpy(tok_p->prod_name, n_license.tok.prod_name,
            tok_p->max_name - n_license.tok.vendor_name);
    }
}

```

明細書の注釈(内容に変更なし)

```

* Function: nl_license
* Description:
*   - This function returns the address of the encode and decode routines.
* Input:
*   - conv_2_ascii_p - storage that indicates whether to conv to ascii
*     code_func_p - storage contains address of encode/decode routines
* Output:
*   - conv_2_ascii_p - return an indication whether the token should
*     be converted to ASCII. It is set to TRUE if it is printable.
*     code_func_p - Contains address of encode/decode routines
* Returns:
*   - Nothing
* Notes:
*   - The reason that the conv_2_ascii_p value is returned is to
*     inform the license production tool if it needs to do
*     ASCII encoding on the license token. The license token must
*     be in ASCII form to allow it to be transmitted over
*     voice phone or to be saved to an end user.
* .....
```

```

void
nl_license(nl_license_p, code_func_p)
    nl_license_p *nl_license_p;
    code_func_p code_func_p;
{
    if (conv_2_ascii_p != NULL)
        *conv_2_ascii_p = TRUE;
    code_func_p->encode_p = nl_encode;
    code_func_p->decode_p = nl_decode;
} /* nl_license */

```

```

* Function: nl_decode
* Description:
*   - Translates a license token from vendor specific format to a
*     client library specific format.
* Input:
*   - tok_p - Pointer to the encoded token
*   - en_length - Length of the encoded token in bytes
* Output:
*   - tok_p - Pointer to the token structure to fill in
*   - client_entry_p - Pointer to the address of the first client usage
*     entry

```

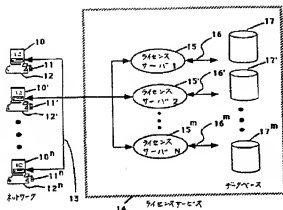



FIG. 1

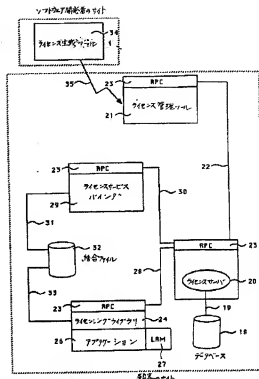


FIG. 2

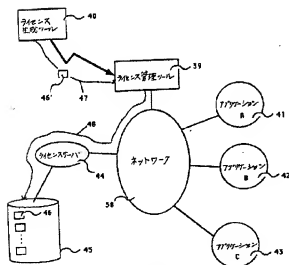


FIG. 3

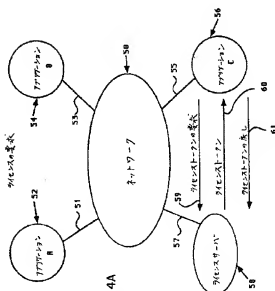
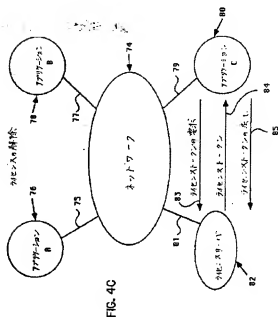
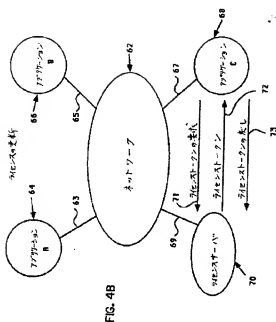


FIG. 4A



手続補正書(方式)

平成 3年 5月16日

特許庁長官殿

1. 事件の表示

平成2年特許願第263242号

2. 発明の名称

コンピュータネットワークシステム内でのソフトウェアの不正使用を防ぐ方法

3. 補正をする者

事件との関係 特許出願人

名称(氏名) サン・マイクロシステムズ・インコーポレーテッド

4. 代理人

居所 東京都千代田区永田町2丁目4番2号
秀和ビル8階
山川国際特許事務所内
☎ (3580) 0961 (代表)

氏名 (6462) 弁護士 山川 政

5. 補正命令の日付 平成 3年 4月16日

6. 補正の対象

(1) 明細書 4P~47P

7. 補正の内容

(1) 明細書の序書(内容に変更なし)

3.5.16 以上

手続補正書(方式)

平成 3年 5月 日

特許庁長官 殿

1. 事件の表示 特願平2-263242号

2. 発明の名称

コンピュータネットワークシステム内でのソフトウェアの不正使用を防ぐ方法

3. 補正をする者

事件との関係 特許出願人

名称 サン・マイクロシステムズ・インコーポレーテッド

4. 代理人

居所 東京都千代田区永田町2丁目4番2号
秀和ビル8階
山川国際特許事務所内
電話 (3580) 0961 (代表)

氏名 (8462) 弁護士 山川 政

5. 補正命令の日付 平成3年10月22日

6. 補正の対象

明細書(第48頁)

7. 補正の内容

明細書の序書(内容に変更なし)

3.11.7

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成11年（1999）4月30日

【公開番号】特開平4-100148

【公開日】平成4年（1992）4月2日

【年通号数】公開特許公報4-1002

【出題番号】特願平2-263242

【國際特許分類第6版】

G06F	9/06	550
	15/00	330

【F I】

G06F	9/06	550 Z
	15/00	330 Z

44-38861-101 101

平成 年 月 日
9.9.4

院野行番官廳

1. 事件の表示 平成24年特別選考第33242号
2. 確立をする者
事件との関係 特許代理人
名称 サン・マイクロシステムズ・インコーポレーテッド
3. 代理人
識別番号 10064451
登録年月 2007年10月
氏名 山川 敬雄
電話番号 03-5580-0981

4. 修正により増加する貸出額の概 算 : 〇

5. 修正时数省期表 明解清

8. 修正新常用月名、號數及序數

7. 賞正の内容 時作選定の報酬を、邦題の通り賞正する。

轉對請求の處理

1. 少なくとも一つのアプリケーション・プログラムを実行する少なくとも一つのコンピュータを含むコンピュータ・ネットワークにおいて、宿記アプリケーション・プログラムの実行許可の発行に関するネットワーク全範囲にわたる情報をさらに増強する、アプリケーション・プログラムをライセンスするための方法であって、

前記アプリケーション・プログラムのライセンス情報を含み、暗号化された形式で記録される第1ライセンス・トークンを、第1ライセンス・サーバによって第1データベースに記憶するステップと、

前記アプリケーション・プログラムを実行するためにライセンスするため、ライセンス・ライブラリのライセンス・ルーチンによって前記第1ライセンス・サーバに要求されるに応じて、第1ライセンス・サーバによって前記アプリケーション・プログラム用の前記の記憶された第1ライセンス・トークンを生成するステップと、

統合された記号化ライセンストークンが特定アプリケーション・プログラムと統合されたアプリケーション・プログラム特有のライセンス・アクセス・モジュールによって管理・管理された後、統合された記号化ライセンストークンに配置されている暗号化ライセンシング情報を、記号化ライセンシング・ルーチンと共に複製アプリケーション・プログラムによって提供されるライセンシング情報と統合させて管理するステップ。

商標ライセンス・ルーチンによる前記機能に応じて、実行のため前記アプリケーション・プログラムをライセンスする入アップと、

商標ライセンス・ルーチンによる前記ライセンシングに応じて、検索された前記登録ライセンス・トークンを更新するステップと、

更新された前記第1ライセンス・トークンが指定アプリケーション・プログラムと結合された前記アプリケーション・プログラム特有のライセンスモデルによって再書き込まれた後に、更新済みの前記第1ライセンス・トークンと、前

第1ライセンス・サーバに接続するソフトウェア・ユーザによって第1データベースに記憶するための複製ライセンシング・ルーチンによって第1ライセンス・サーバに接続するソフトウェア・ユーザと第1データベースとの間の通信を制御する。

2. 自己製1ライセンス・トークンを搭載アプリケーション・プログラムの発生から受け取り、登記簿1ライセンス・トークンを前記第1データベースに記憶するため前記第1ライセンス・サーバに供給するスタッフをさらに含む。

前記第1ライセンス・トークンは、前記アプリケーション・プログラムの配布媒体から受領され、暗号化された形式でライセンス管理ツールによって前記第1ライセンス・サーバに送信されることを特徴とする。

請求項1に記載の方法。

8. 販売アプリケーション・プログラムの配布媒体からの、前記アプリケーション・プログラム特有のライセンス・アクセス・モジュールと統合され、前記コンピュータにインストールされた、複製アプリケーション・プログラムを受発するスタッフをさらに含む。

前記アプリケーション・プログラム特有のライセンス・アクセス・モジュールが、前記アプリケーション・プログラムと統合され、前記アプリケーション・プログラムの売主によって前記記憶媒体に固められていることを特徴とする。

読家様！に世間の方法。

4. 開発アプリケーション・プログラムを実行するためライセンスを許可にするため、ライセンシング・ライブラリのライセンスを許可するものによって前記第1ライセンス・サーバに要求されるものに応じて、開発アプリケーション・プログラム用の前記の配信された第1ライセンス・トークンを前記第1ライセンス・サーバによって提供するステップ。

複製された音源第3ライセンス・トークンが前記アプリケーション・プログラムと統合されたアプリケーション・プログラム特有のライセンス・アクセス・モジュールによって管理解読される際に、複製された音源第3ライセンス・トークンに記述されている前記ライセンシング情報を、前記ライセンス許可サーバーと向い複製アプリケーション・プログラムによって照合されるライセンス情報欄と宛を含めて送信するステップと。

複製ライセンス不許可ルーチンによる複製検出に応じて、実行のため複製アプリケーション・プログラムをライセンス不許可にするステップと、

れた著作権・ライセンス・トランザクションを構築するプラットフォーム

実務者の登記簿「ライセンス・トークン」が自己アプリケーション・プログラムと結合された複製アプリケーション・プログラム特有のライセンスモジュールによって再暗号化された後に、更新済みの複製簿「ライセンス・トークン」を、前記簿「データベース」に記憶するため前記ライセンス不許可ルーチンによって前記簿「ライセンス・サーバ」に書き出すステップと

5. 商品アプリケーション

れた形式で記述される第2ライセンス・トークンを、第3ライセンス・サーバによって第2データベースと照合するステップと。

前記ライセンスリング、ライブツリーを読み込みファイルにアクセスして、前記第1ライセンス、サーバを選択し前記第1ライセンス、トークンを使用するように前記の第1ライセンス、サーバと第2ライセンス、サーバを選択する第1ライセンス、サーバ識別子と第2ライセンス、サーバ識別子を読み込みファイルに記憶するステップ。

前記の第1ライセンス・サーバと第2ライセンス・サーバを識別して、前記の第1ライセンス・サーバ識別子と第2ライセンス・サーバ識別子をライセンス・サービス・バインディングによって前記の結び込みファイルに転送するステップとをさらに含む処理フロー1に示する方法。

8. 少なくとも一つのアプリケーション・プログラムを実行する少なくとも一つのコンピュータを含むコンピュータ・ネットワークにおいて、宿居アプリケーション・プログラムの継ぎ手の実行に対するネットワーク全範囲にわたる保護をさらに増強する、アプリケーション・プログラムをライセンスするための装置であって、

自製アプリケーション・プログラムのライセンス料預納を含め、暗号化された形式で提供される第1ライセンス・トークンと暗号化するための第2ライセンス・

又を含む第1データベース手段と、

前記第1ライセンス・トークンを官定第1データベースに伝送し、かつ記憶された前記第1ライセンス・トークンを官定第1データベースから検索するための第1ライセンス・サーバを含む、官定第1コンピュータと前記第1データベースとに結合された第1ライセンス・サーバと、

[illegible]

目録Aアプリケーション・プログラム特定の、目録Aアプリケーション・プログラムと結合された、後述された前記第1ライセンス・トークンを暗号解読し、かつ更新済みの前記第2ライセンス・トークンと再編成化するための、前記コンピュータによって実行されるライセンス・アクセス・モジュールを含む装置。

7. 有償アプリケーション・プログラムの販売価格から固定第1ライセンス・トラングを費材に差し引き、前表第1ライセンス・トラングを固定第1データベースに配備するための第2第1ライセンス・サーバに算出するための、固定第1ライセンス・サーバ・サーバに結合されたライセンス管理ツールが利用される。固定第1ライセンス・トラングが第2第1ライセンス管理ツールによって受領され、暗号化された形式で固定第1ライセンス・サーバに供給されることを意味する。前表第1に記述の通り。

5. 前記アプリケーション・プログラムのライセンス情報を含み、暗号化された形式で記憶されている、第2ライセンス・トークンを記憶するための第2データベースを含む第2データベース。

第2ライセンス・トークンを第2データベースに授け、かつ拒絶された第2ライセンス・トークンを第2データベースから検索するための第2ライセンス・サーバを含む、第2コンピュータと第2データベースとを結合させた第2ライセンス・サーバ単位と

前記ライセンシング・ライブラリが属するファイルにアクセスして、其内容
のライセンス・サーバを選択し前記第1ライセンス・トークンを検索するように、
前記の第1ライセンス・サーバと第2ライセンス・サーバを照合する第1ライ
センス・サーバ識別子と第2ライセンス・サーバ識別子が実装するための。属する
ファイルにアクセスして、其内容のライセンス・サーバを選択し前記第1ライ

図 6-11 ライセンス・サーバの構成と、第 1 ライセンス・サーバと第 2 ライセンス・サーバの接続関係